10/815524
12-06-04

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 :
G06F 13/00

A1

(11) International Publication Number: WO 00/67132

(43) International Publication Date: 9 November 2000 (09.11.00)

(21) International Application Number: PCT/US00/11394

(22) International Filing Date: 28 April 2000 (28.04.00)

(30) Priority Data:

60/131,793	30 April 1999 (30.04.99)	US
60/134,883	19 May 1999 (19.05.99)	US

(71) Applicant (*for all designated States except US*): CENTENNIAL TECHNOLOGIES, INC. [US/US]; 7 Lopez Road, Wilmington, DE 01887 (US).

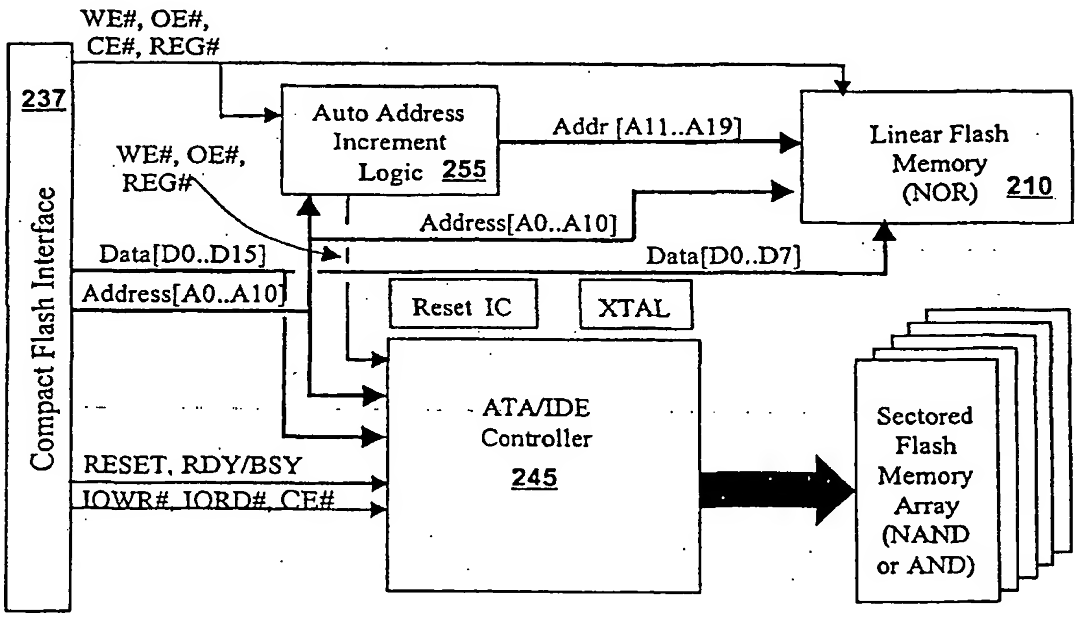
(71)(72) Applicants and Inventors: LAMBERT, Grady [US/US]; 2045 Ashbury Drive, Clearwater, FL 33764 (US). HENDRICKSEN, Craig [US/US]; 72 Main Street, Framingham, MA 01702 (US).

(74) Agent: LYON & LYON LLP; Suite 4700, 633 West 5th Street, Los Angeles, CA 90071 (US).

(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: COMBINATION ATA/LINEAR FLASH MEMORY DEVICE



(57) Abstract

A single combination data storage device (140) provides both firmware (210) and disk emulation storage (225) on a single removable media device. Permanent and programmable data of the firmware can be modified on a support computer making the combination device useful for upgrading and initially configuring the firmware for embedded systems as well as their applications, OS kernel, and user data. In a preferred embodiment, the device is implemented with a combination of flash memory (210) for firmware and ATA/flash (225) providing drive emulation in a PC Card or other standard form factor.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

COMBINATION ATA/LINEAR FLASH MEMORY DEVICE

SPECIFICATION

Field of the Invention

The present invention relates to data storage devices for computers and more particularly to computers with removable data storage devices that store both the computer's boot firmware and other permanent data, normally associated with hard disks, on the single removable data storage device.

Background of the Invention

Referring to Figs. 1 and 2, generally speaking, desktop and laptop computers have traditionally been built around a consistent model. A processor 10 is programmed to address a non-volatile memory 20 that stores a basic operating system (not shown). The basic operating system contains the intelligence necessary for the computer to image (copy) a full operating system into a volatile random-access memory (RAM) 25 from a medium that typically has a non-volatile, but slower read/write performance. Examples of such permanent media are hard disks, floppy disks, removable or erasable non-volatile memory (e.g., flash), or any suitable I/O device 30. More technically, although not required to understand the present specification, the basic operating system generally contains a vector table, basic input output system (BIOS), and a boot loader. That is the basic model: large complex and ephemeral data is stored on a hard or floppy drive, and the permanent less-frequently modified software is stored on a hard-wired basic operating system memory. This basic model applies, and increasingly so, to embedded systems such as cash registers, palmtop PCs, and dataloggers.

One technology that permits the general computer model to be applied to compact portable embedded systems and computers permits the use of a type of solid-state memory to take the place of larger more vulnerable hard drives. One example is called ATA/IDE Flash memory. It employs flash memory with an interface designed to permit its data to be addressed as data from a hard drive according to the standard ANSI

ATA/IDE standard. In other words, the Flash memory appears to the rest of the computer to be a hard drive. The ATA Flash memory can be loaded with new data for upgrading applications or operating system application programming interfaces APIs just as in the traditional model. So, even for systems that include a Flash memory to physically replace the hard drive, the systems are built around the old conventional model of a small basic operating system memory and a separate larger storage device. ATA/IDE drives offer the flexibility of being cross platform compatible.

The boot software stored on the non-volatile memory 20, is usually peculiar to the particular hardware to which it is connected. Thus the non-volatile memory is often connected directly to the computer (e.g., soldered to the motherboard) with no convenient mechanism for modifying it. The large permanent storage (represented generally by I/O devices 30 in the figure) must, however, be conveniently addressable by users to allow modification or upgrade of applications and operating system elements. The separation of all but the bare essential elements for operating the host computer from the rest of the operating system and applications stored on a hard drive also provides the ability to recover from a corrupted hard drive. By following a straightforward recovery procedure involving booting an operating system from a removable medium, backed up data can be restored to the old or a new hard drive. Even so, firmware occasionally must be updated and in most systems, this requires the attention of a technician.

ROM Shadowing is a process by which the contents of the ROM boot image are copied from ROM to onboard DRAM or SRAM to speed access times and enhance the performance of the system. In some cases, system designers can utilize the relatively fast access times of Linear Flash directly from the flash memory, without the need to shadow the code in RAM, (a technique called execute-in-place or XIP) thus avoiding the copying process and reducing RAM requirements.

Firmware memories sometimes need to be changed just like applications, but modifying the basic operating system presents some challenges. The basic operating system is normally stored in a stand-alone memory device (EPROM, EEPROM, Flash memory, Mask ROM, etc.) that is not positioned or connected to the system to permit ready access for replacement or reprogramming. For example, in desktop systems, it is usually soldered to the motherboard and is accessible for reprogramming (outside of a few user-accessible parameters through a user interface (UI) available at bootup) only by removing the cover from the computer and connecting to it through a specialized operation. When an upgrade must be made to the firmware, it is usually necessary for a technician to do it because of the need to address the peculiar requirements of the device.

Summary of the Invention

A combination ("combined device") furnishes all of the necessary read-only (ROM) memory resources, or ROM-like memory resources, typically required for a PC or embedded system combined with a mass storage device in a single removable module. In a preferred embodiment, the form factor of the device is that of the PC Card (PCMCIA) or CompactFlash® card (CF® card). The combined device combines the storage and interface for the two traditionally segregated types of non-volatile data resources. The first type is used to store the computer's firmware such as vector table, basic IO system (BIOS), and the boot loader (hereafter referred to as the *boot image*). The second type is used to store the operating system (OS) kernel, software applications, and user data. The second type is that normally associated with ATA/IDE drives and in a preferred embodiment, with ATA flash memory devices which are flash memories controlled to emulate a drive.

In a preferred embodiment, the combined device communicates with a host computer through an ATA/IDE interface. The ATA/IDE interface specification provides for three fundamental interface protocols for communication: (1) memory mapped, (2) I/O mapped and (3) True IDE. The I/O mapped protocol comprises three sub-modes for I/O mapped data transfers to and from the card (i.e., Contiguous, Primary and Secondary), which will be considered a single entity for the purposes of this disclosure. In this embodiment, the host computer, which may be a PC, an embedded system, or other device, loads (for a traditional model where the firmware is imaged into RAM) or reads data (for an execute-in-place model) firmware through one of the channels available through the ATA/IDE interface. The channel or mode allocated for accessing the firmware or boot image may be any of the three available. When the firmware must be upgraded, the combined device is removed from the host computer and connected to a support computer that downloads new firmware to the combined device.

By incorporating both the firmware and other permanent storage on the same removable combined device, firmware can be updated easily. The combined device allows the user to remove, replace, modify, or upgrade both memory resources by simply removing, the combined device from the host and plugging it into a similar adapter on the support computer. In one embodiment, the support computer is envisioned to be desktop computer and the system that boots from the combined device is an embedded system device. The support computer writes new data to the combined device. The new data may be obtained, for example, from a network or media supplied to users from the embedded system manufacturer. Once the new firmware is loaded on the combined device, it is reconnected to the host computer and the host computer booted normally.

In many embedded system applications, data retrieved by the embedded system may routinely be downloaded from, or uploaded to, the mass storage of the embedded system device. A convenient way to do this is for the mass storage devices to be collected from the embedded system users and read (or written to) by the support computer. For example, data downloaded could be data collected via the host data terminal (i.e., data logger). During such read and/or write operations with mass storage, a background operation may be performed by the software application used to download (upload) the data. That is, the software application may, at the same time, also verify the revision level of the boot image stored on the linear flash memory device within the combined card. This is one of the advantages of placing the firmware on the same removable card as the embedded system device.

In an alternative embodiment, the firmware and other data storage (e.g., ATA/linear flash memory) device are provided on separate portions of a single device. A main adapter section, or "shuttle," may contain one of the memory elements (for example, the firmware portion) and the other, the piggy-back portion (e.g., a CF® module, may contain the other memory element (for example, the ATA linear flash memory that emulates a hard drive). The interconnection may be via a standard interface that allows the piggyback portion to be connected to a support computer via that interface. One example of such an interface is CF® with a CF®-to-PCMCIA shuttle adapter.

The convenience of the combined device is apparent in a scenario in which many embedded systems are owned and maintained by the same entity, for example, a retailer with a large number of bar-code scanners. To upgrade all of these in the traditional way involves a great deal of technical service since the embedded system's firmware has to be reprogrammed or replaced through a specialized operation as discussed in the background section. With the combined device, the owner of the equipment can make all of the firmware changes quickly and conveniently. Each combined device can be removed from the corresponding embedded system device, a bar-code scanner, and inserted into the support computer, upgraded, and replaced in the embedded system device. The upgrade operation is controllable by the software provider and can be made very simple since no user input is required save that of initiating the operation.

Another benefit of the invention is apparent in the above scenario. In situations where a single entity is responsible for setting specific parameters of the firmware for a large number of devices, it can be very inconvenient to set those parameters from the individual device interfaces. This may require many manual setup operations, one for each embedded system device. Using the invention, the responsible technician can collect the combined devices from all the systems requiring programming and run a program on a support computer to change the data settings in the firmware aboard the combined device.

Still another benefit of the invention is apparent when the host computer is damaged or can no longer be used. In such an instance, all of the non-volatile storage can be quickly removed from the non-functioning device and immediately inserted into a similar terminal. In the flash memory embodiment, where firmware and other (applications, OS kernel, etc.) software are stored in, for example, a flash ROM and ATA/IDE flash, all customized parameters set in either set of data are preserved. For example, the BIOS settings that are already stored in the combined device are installed into the replacement hardware with the installation of the combined device from the non-functioning embedded system. For execute in place configurations, where the embedded system runs code and reads data directly from non-volatile storage rather than imaging into RAM, the full state of the embedded system can be restored after a crash. That is, user data, open files, etc., at the time of malfunction, can be fully reinstated.

Still another context in which the invention has advantages is where the user's current configuration and software are to be transferred to a new machine. In a preferred embodiment, the combined device is connected through a standardized interface. Upgraded or new embedded system hardware can accept the combined device from the previous system so that the embedded system can be placed back online with all of its previous settings without reprogramming.

Any customization or configuration parameter preferences would be transitioned to the replacement platform, resulting in a considerable reduction in the time required to get the unit operational again. In fact, because the unit may contain all the "intelligence" of the state of the computer at the time of malfunction, the combined device provides a fast track to reproducing the lost state,. Secondly, as OS and Application Software upgrades and revisions are made, so too can the BIOS be upgraded. Simply reprogramming the Linear Flash section of the integrated memory module results in, in-effect, a new BIOS (or vector table, or boot loader, etc.) The time and expense incurred in maintaining configuration control of field-deployed platforms and the usually complex nature of upgrading the embedded soldered-down memory device used to store the BIOS are alleviated.

In a preferred embodiment, the combined device combines a small amount of linear flash (e.g., about 1MB) with a large quantity (e.g., 8MB or more) of solid-state ATA/IDE flash in a single module. The module would be useful to OEM manufacturers of embedded systems such as portable handheld devices. Such manufacturers would design their systems around the combined device so that firmware is read through the appropriate interface mode. Packaging form-factors can be, but are not limited to, PC card (PCMCIA), CF®, and single & dual in-line memory modules (SIMM & DIMM). Both flash technologies, linear and ATA/IDE, share the same address and data bus, dedicated control signals are used by the host to differentiate between the two memory resources. .

According to an embodiment, the invention provides a data storage device for interfacing with a host computer via a data interface. The host computer is configured to read firmware including a basic operating system (firmware), from the data storage through the interface rather than through the usual mechanism of an on-board EPROM chip. The removable data storage device has a first memory device adapted to store the firmware. The removable data storage device also has a second memory device adapted to store data other than the firmware. For example, the usual data stored on a hard drive is stored in this second memory device. The data storage device has a physical data channel adapted to connect with a physical data channel of the interface. The first and second memory devices share resources of the physical data channel. At least one of the first and second memory devices is configured to employ the physical channel only in the presence of a signal provided through the interface by the host computer so the two memory devices do not collide. Optionally, the first memory device includes a linear flash memory. Also, the second memory device may include a sectored flash memory and a controller programmed to provide ATA/IDE disk emulation. The physical data channel may include a PC Card CF®, SIMM, DIMM or other removable module interface and, in combination with this feature, the first memory device may be a linear flash memory.

According to another embodiment, the invention provides a computer with a peripheral interface for communicating with a connected removable data storage device. the computer has a controller programmed to address the removable storage device and read basic operating system program from it. The basic operating system is the basic steps and data normally associated with a BIOS chip. That is, this data enables a boot operation of the computer. The peripheral interface may include a PC Card adapter. The peripheral interface may be configured to permit a repeated connection and disconnection of the removable storage device.

According yet another embodiment, the invention provides a memory card with a physical data communications interface adapted to permit repeated connection and disconnection to and from a host computer via a plug-in adapter. It also has a first non-volatile memory device with a controller programmed to emulate a mass storage device of a host computer and a second non-volatile memory device storing a bootstrap program for a host computer. The second non-volatile memory device share physical resources of the communications interface with the first non-volatile memory device. The physical data channel may include a PC card adapter. The first non-volatile memory device may be programmed to emulate an ATA/IDE specification disk drive. The second non-volatile memory device may include an EPROM, an EEPROM, Mask ROM, or a linear flash memory.

According to yet another embodiment, the invention provides a PC card with a sectored flash memory. The card has a communications interface adapted to provide ATA/IDE disk emulation between a host computer and the sectored flash memory. It also has a linear flash memory selectively sharing a physical channel of the communications interface with the sectored flash memory so a host computer may selectively address the linear flash memory or the sectored flash memory.

According to yet another embodiment, the invention provides a removable storage device for interfacing via a data interface with a host computer. The host computer is configured to read boot data, including a basic operating system, through the interface. The removable storage device has a first memory device adapted to store the firmware and a second memory device adapted to store data other than the firmware. The removable storage device share a physical data channel adapted to connect with a physical data channel of the interface. The first and second memory devices share resources of the physical data channel. At least one of the first and second memory devices is configured to employ the physical channel only in the presence of a signal provided through the interface by the host computer, so a collision between the two memory devices is avoided.

While the invention will now be described in connection with certain preferred embodiments and examples and in reference to the appended figures, the described embodiments are not intended to limit the invention to these particular embodiments. On the contrary, it is intended to cover all alternatives, modifications, and equivalents as may be included within the scope of the invention as defined by the appended claims. Thus, the following description and examples of the preferred embodiments of the invention are only intended to illustrate the practice of the present invention. The particular embodiments are shown by way of example and for purposes of illustrative discussion of the preferred embodiments of the present invention.

The particular embodiments are presented in the cause of providing what is believed to be the most useful and readily understood description of the principles and conceptual aspects of the invention. In this regard, no attempt is made to show structural details of the invention in more detail than is necessary for a fundamental understanding of the invention. The description, taken with the drawings, makes it apparent to those skilled in the art how the several forms of the invention may be embodied in practice.

Brief Description of the Drawings

Fig. 1 illustrates a basic embedded computer system according to the prior art.

Fig. 2 illustrates a typical memory map that would be found in an embedded system, the address ranges and partition information are given solely for the purposes of this example, are not intended to reflect the internal architecture of any specific micro-controller or micro-processor.

Fig. 3 is a functional block diagram of a general embodiment of the invention.

Fig. 4 is a functional block diagram of an embodiment of the invention that employs a linear flash for firmware storage, a sectored flash memory with an ATA controller for the mass storage, a PCMCIA interface for communication with a host computer.

Fig. 5 is a functional block diagram of an embodiment of the invention that employs a linear flash for firmware storage, a sectored flash memory with an ATA controller for the mass storage, a CF® interface for communication with a host computer.

Detailed Description of the Illustrated Embodiments

Referring to Fig. 3, a host computer 45 has a CPU 10 that addresses a RAM 25 and an I/O device 140 through an I/O interface 135. The host computer may be any type of computer, although the invention appears to be most advantageously applied to embedded system devices such as bar-code scanners, dataloggers, and various specialized portable computers as well as palmtops, organizers, personal digital assistants, digital control systems, etc. The I/O interface can be SIMM, DIMM, PC Card (PCMCIA), CF®, ATA/IDE, or any other type of interface.

The I/O interface 135 provides the physical interconnections between the removable combined device 140 and the host computer 45. Some or all of the signal interconnections that make up the I/O interface 135 provide connection to a storage device used to store the firmware 110, for example, EPROM, EEPROM, or flash memory. Some or all of the signal interconnections that make up the I/O interface 135 provide connection to a storage device used to store the other data 125, for example, an ATA/IDE flash memory that emulates an IDE hard disk. The interconnection between the I/O interface and the storage device 110 and/or 125 may require a controller 145a and 145b, respectively, intervening between the I/O interface and the storage device 110 and/or 125.

Referring now to Fig. 4, in an embodiment of the device of Fig. 3, a flash memory 210 is used for storage device 110 and an ADA/IDE flash memory device 225 is used for a storage device 125. An ATA/IDE controller intervenes between the PC card I/O ~~interface 235 and sectored flash memory 225 used to emulate an ATA/IDE device, like a~~ hard disk. Essentially no control logic, outside of the write-enable, output enable, and card enable lines, is required to provide addressing and data connections between the host computer and the linear flash memory 210. These lines are shared with those used by the ATA/IDE flash memory device 250. The host computer boots from the linear flash memory 210 using a certain set of signal lines available in the I/O interface (e.g., a PC card interface). After the boot operation is finished (assuming the required parts of the firmware have been imaged in memory), these signal lines can be relinquished for use by the ATA/IDE device 250.

The embodiment of Fig. 4 contains an ATA/IDE flash memory device 250 with a linear flash portion 210 that selectively or permanently interfaces directly with the same interface used to enable the ATA/IDE disk drive emulation. In other words, the card generally functions as an ATA/IDE flash card except when and where access to the firmware data is required for either reading or writing operations. Generally, in the host computer, the firmware in the linear flash is not written over. It is only when the firmware is to be upgraded that the linear flash part of the device is written to. This is ordinarily when the device is temporarily connected to a support computer for reprogramming, parameter-setting, or upgrading of the firmware.

In accordance with the PC card ATA specification, ATA/IDE cards can be configured to operate in three basic operational modes. Additionally, as prescribed in the specification, PC Card compliant cards shall provide for an attribute memory plane for the purposes of allowing the host to identify the type and operational characteristics of the card. The interface control signals necessary for transferring data to and from the attribute memory plane and the three modes of operation are shown in Table 1.

Table 1. Read/Write Access Modes for ATA/IDE Flash Cards.

Mode	REG#	CE1#	CE2#	IORD#	IOWR#	OE#	WE#
Attribute Memory (Read)	L	L	H	H	H	L	H
(Write)	L	L	H	H	H	H	L
Memory Mapped (Read)	H	L	L	H	H	L	H
(Write)	H	L	L	H	H	H	L
I/O Mapped (Read)	L	L	L/H	L	H	H	H
(Write)	L	L	L/H	H	L	H	H
True IDE (Read)	X	L	H	L	H	L	H
(Write)	X	L	H	H	L	L	H

Notes:

1. Combinations of CE1# and CE2# select Byte (8-bit) or Word (16-bit) accesses.
2. Consult ATA/IDE specification for detailed description of these signals.

As illustrated the linear flash and the ATA/IDE flash memories may be treated as two separate devices. Data transfers to and from the linear flash memory are initiated in the same manner as a standard 8-bit flash card. write enable (WE#) and output enable (OE#) coupled with card enable (CE1#) strobe data transfers to and from the combined device. Both address and data bus signals are shared by the linear 210 and ATA flash memory sections. The mapping of the memory resources by WE# and OE# are for purposes of illustration only and other alternatives are possible. Some applications may require the use of WE# and OE# to support IO transfers to and from the ATA/IDE Flash, in this case, the IO mapped resources of IOWR# and IORD# or any other available signal could be used to support data transfers to and from the linear flash device 210.

Exemplary Implementation

As stated earlier, the multi-function aspects of the ATA/IDE PC Card and the CompactFlash® CF® interface specifications permit the maintenance of compliance and at the same time accommodate the additional functionality of linear flash memory. One strategy is to mask one of the four ATA/IDE operational modes and use the freed interface resources, or "channel," to address the linear flash memory device 210.

With respect to linear flash memory, the CF® interface imposes a physical bandwidth limitation for addressing the linear flash. The CF® interface consists of 50 pins, of which address lines, A0-A10 are supported. To address 1MB of linear memory space, one would need to generate A0-A19 address lines, all the while maintaining compliance with the functional requirements specified with the CF® standard. To overcome the I/O bandwidth limitation, a paging scheme is employed to allow addressing beyond the CF® addressing limitation. Shown in Fig. 5, is a block diagram rendering of the combined device using the CF® I/O interface 237.

Implementation based on the CF® I/O interface standard requires that certain rules be followed in accessing the physical memory of the linear flash device (i.e., Start Address = 0x0000h and End Address = 0xFFFFFh.) According to an auto address increment logic 255, the host follows some basic protocol rules with respect to write, read, and erase operations. A brief description of one example protocol as it applies to write, read, and erase operations is given below.

2.4.1 CF WRITE Operations

In one embodiment, write cycles are only supported for addresses beginning at address 0x0000h and incrementing sequentially to 0xFFFFFh, random writes to any location(s), in arbitrary order, are not supported. Since in any envisioned application, the linear flash would be addressed for writing only when a boot image is to be written, this requirement is expected to have virtually no impact on the utility of the combined device.

Alternatively, a write operation to an arbitrary address location is possible. However, the operation will require a three-bus write cycle sequence. The first write bus cycle consists of a write to the internal page register, this register is accessed via assertion of the register line, REG#. Once the appropriate 2K page has been selected, any subsequent write operations to addresses within the current 2K page will be carried out as specified by the operational requirements of the specific linear flash device (e.g., Intel® versus AMD® programming algorithms).

2.4.2 CF READ Operations

To permit read cycles beyond the address space of the CFA interface, the host accesses the linear flash data in a sequential manner beginning at address 0x0000h. In most embedded applications, the boot image, located at address 0x0000h is read by the CPU on either power-up or reset, and is typically read in a sequential address manner so, again, this restriction is not regarded as onerous.

Alternatively, to read from a specific address within the linear memory device, the host must first write a page number to the page register by asserting the page register, then sequential read operations can resume from this new location.

2.4.3 CF ERASE Operations

The block erase organization of Intel Series II flash chips is realized by requiring the host to write the erase block command to the first 64kb block, the host then polls the status register to detect a successful erase operation.

Alternatively, to perform a block erase, the host must first write to the page register, selecting a 2kb page number within the 64kb block to be erased before the write operation. Then, for example, the standard Intel Series II flash Block Erase algorithm can be implemented.

Various other ways of overcoming the address bandwidth limitation may also be employed. For example, the narrow-band address applied to the auto address increment logic 255 could be multiplied by that device by some factor to provide a the start address of a block. Then the auto address increment logic would generate a sequence of addresses successively applied at the broad-band address lines to sequentially retrieve the block of data. That is, the narrow band address applied across A0-A10 represents a start address divided by the block size for a block containing the datum required.

It will be evident to those skilled in the art that the invention is not limited to the details of the foregoing illustrative embodiments, and that the present invention may be embodied in other specific forms without departing from the spirit or essential attributes thereof. The present embodiments are therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description, and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

We claim:

1. A removable data storage device for interfacing via a data interface with a host computer, the host computer being configured to read boot data, including a basic operating system, through said interface, said removable data storage device comprising:

a first memory device adapted to store said firmware;

a second memory device adapted to store data other than said firmware;

a physical data channel adapted to connect with a physical data channel of said interface;

said first and second memory devices sharing resources of said physical data channel;

at least one of said first and second memory devices being configured to employ said physical channel only in the presence of a signal provided through said interface by said host computer, whereby a collision between said two memory devices is avoided.

2. A device as in claim 1, wherein said first memory device includes a linear flash memory.

3. A device as in claim 1, wherein said second memory device includes a sectored flash memory and a controller programmed to provide ATA/IDE disk emulation.

4. A device as in claim 1, wherein said physical data channel includes a PC Card CF®, SIMM, DIMM or other removable module interface.

5. A device as in claim 4, wherein said first memory device is a linear flash memory.

6. A computer, comprising:

a peripheral interface for communicating through said interface with a connected removable data storage device;

a controller configured to address said removable storage device and read basic operating system program therefrom, whereby said a boot operation of said computer is enabled.

7. A computer as in claim 6, wherein said peripheral interface includes a PC Card adapter.

8. A computer as in claim 6, wherein said peripheral interface is configured to permit a repeated connection and disconnection of the removable storage device.

9. A memory card, comprising:

a physical data communications interface adapted to permit repeated connection and disconnection to and from a host computer via a plug-in connection;

a first non-volatile memory device with a controller programmed to emulate a mass storage device of a host computer; and

a second non-volatile memory device storing a bootstrap program for a host computer, said second non-volatile memory device sharing physical resources of said communications interface with said first non-volatile memory device.

10. A card as in claim 9, wherein said physical data channel includes a PC card adapter.

11. A card as in claim 9, wherein said first non-volatile memory device is programmed to emulate an ATA/IDE specification disk drive.

12. A card as in claim 9, wherein said second non-volatile memory device includes one of an EPROM, an EEPROM, Mask ROM and a linear flash memory.

13. A PC card, comprising:

sectored flash memory;

a communications interface adapted to provide ATA/IDE disk emulation between a host computer and said sectored flash memory;

a linear flash memory selectively sharing a physical channel of said communications interface with said sectored flash memory, whereby a host computer may selectively address said linear flash memory or said sectored flash memory.

14. A removable storage device for interfacing via a data interface with a host computer, the host computer being configured to read boot data, including a basic operating system, through said interface, said removable storage device comprising:

a first memory device adapted to store said firmware;

a second memory device adapted to store data other than said firmware;

a physical data channel adapted to connect with a physical data channel of said interface;

said first and second memory devices sharing resources of said physical data channel;

at least one of said first and second memory devices being configured to employ said physical channel only in the presence of a signal provided through said interface by said host computer, whereby a collision between said two memory devices is avoided.

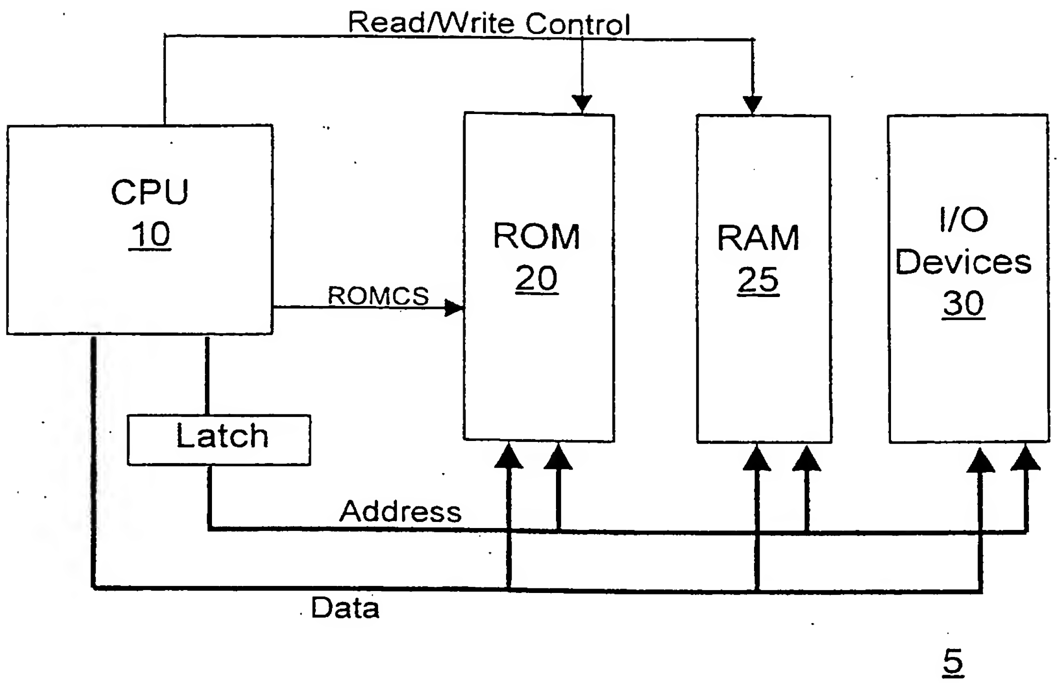
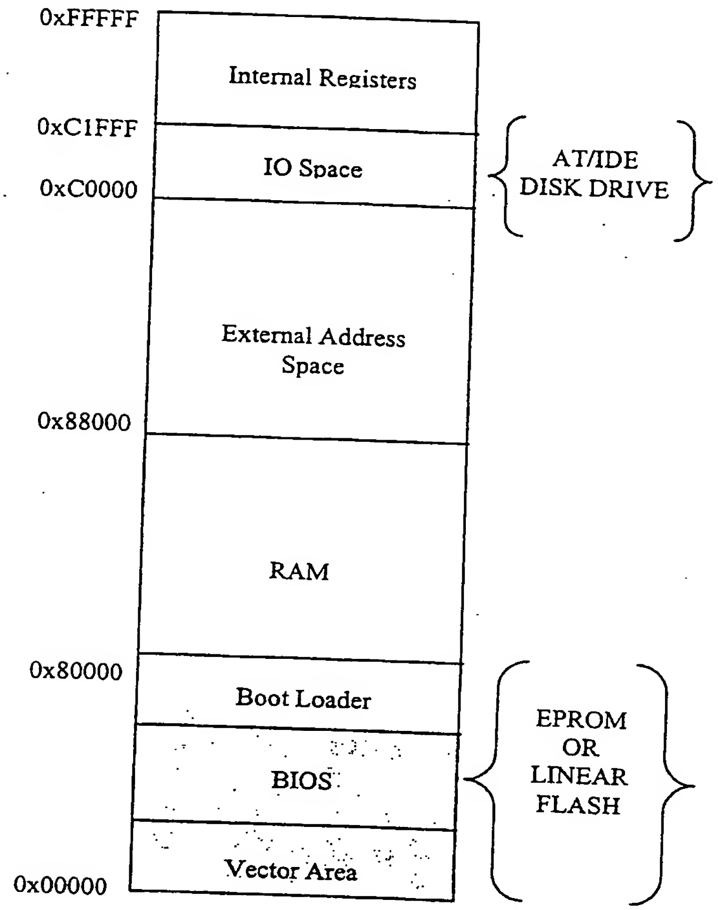

Read/Write Control
CPU
10
ROMCS
ROM
20
RAM
25
I/O
Devices
30
Latch
Address
Data
5

Fig. 1



System Memory Map

Fig. 2

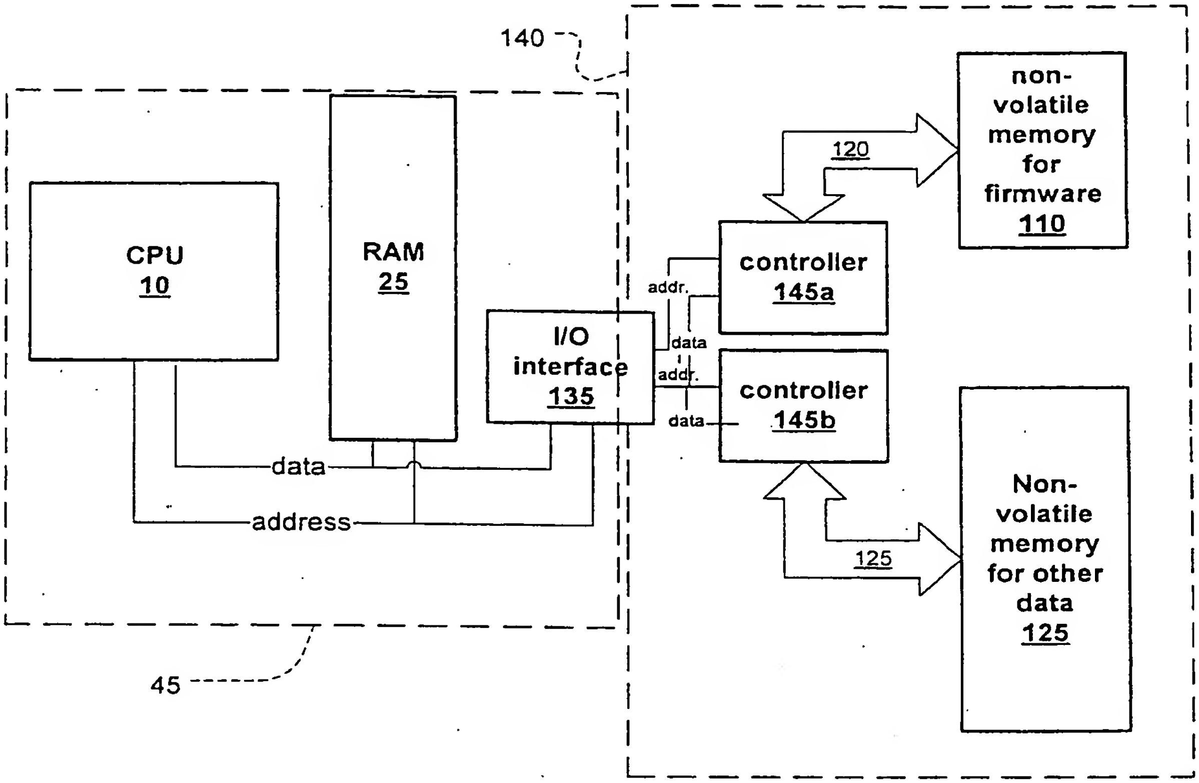

140
CPU
10
RAM
25
I/O
interface
135
data
address
45
addr.
data
addr.
data
controller
145a
controller
145b
120
125
non-
volatile
memory
for
firmware
110
Non-
volatile
memory
for other
data
125

Fig. 3

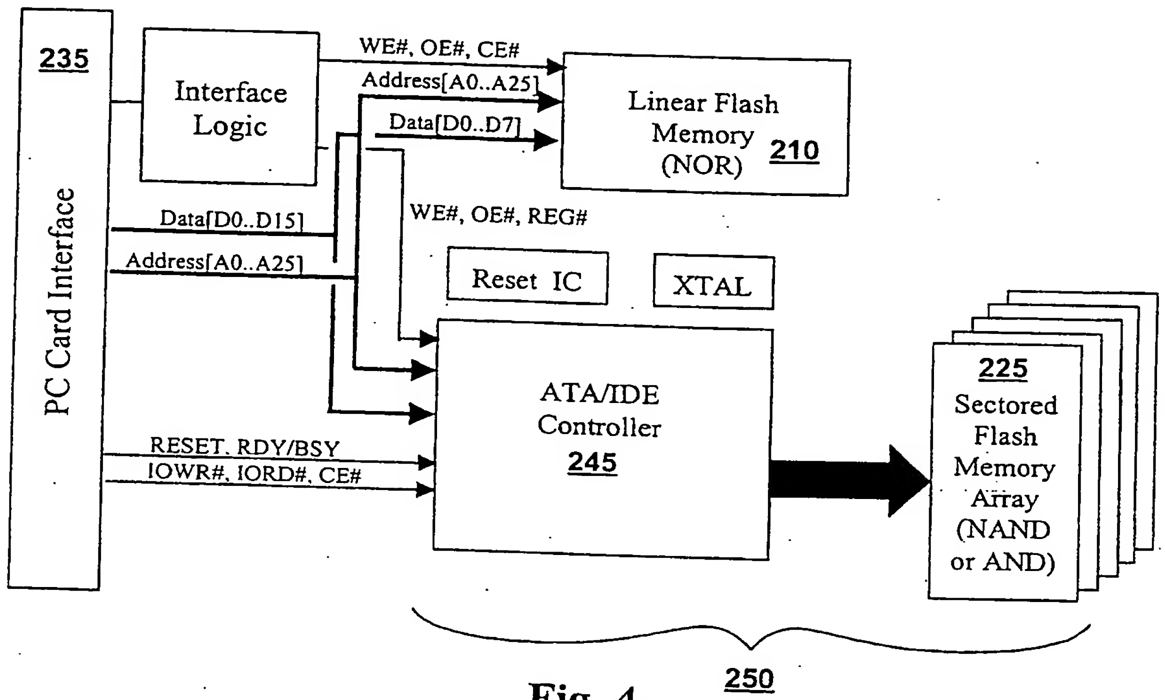

235
PC Card Interface
Interface Logic
WE#, OE#, CE#
Address[A0..A25]
Data[D0..D7]
Linear Flash Memory (NOR) 210
Data[D0..D15]
Address[A0..A25]
WE#, OE#, REG#
Reset IC
XTAL
ATA/IDE Controller 245
RESET, RDY/BSY
IOWR#, IORD#, CE#
225
Sectored Flash Memory Array (NAND or AND)
250

Fig. 4

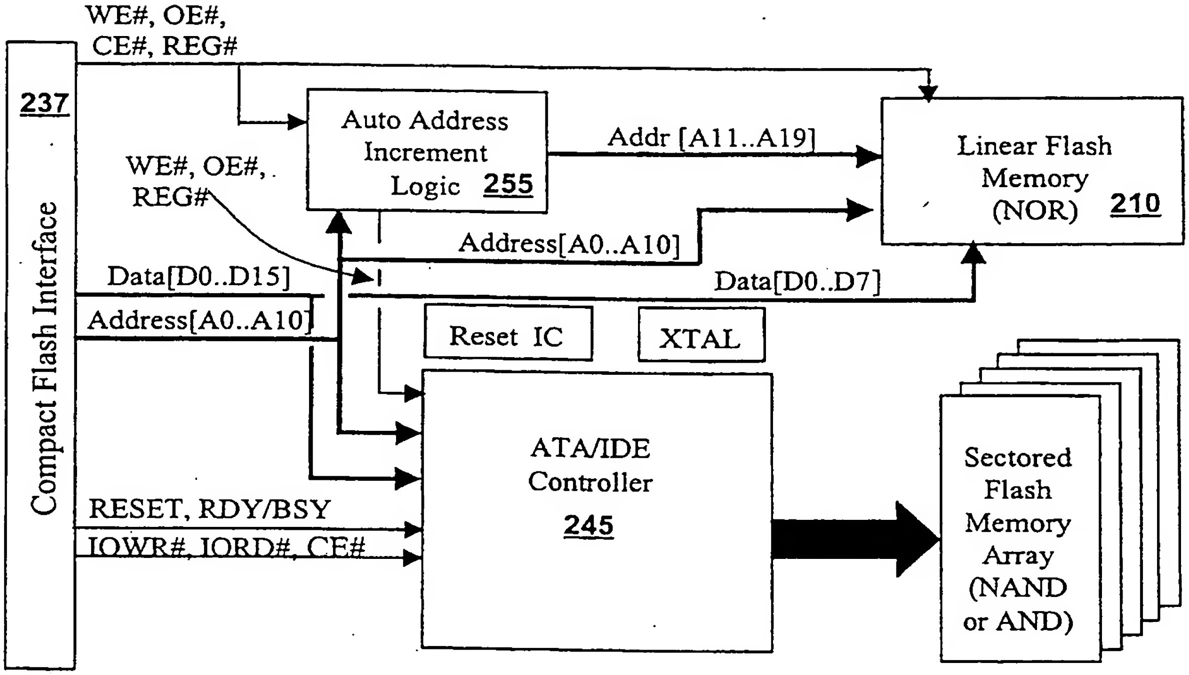

WE#, OE#,
CE#, REG#
237
Compact Flash Interface
Auto Address
Increment
Logic 255
Addr [A11..A19]
Linear Flash
Memory
(NOR) 210
WE#, OE#,
REG#
Address[A0..A10]
Data[D0..D15]
Data[D0..D7]
Address[A0..A10]
Reset IC
XTAL
ATA/IDE
Controller
245
Sectored
Flash
Memory
Array
(NAND
or AND)
RESET, RDY/BSY
IOWR#, IORD#, CE#

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/11394

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 13/00
US CL :711/115, 103; 710/13, 102
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 711/115, 103; 710/13,102

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST
removable, memory, firmware, flash, emulation, disk, pc, card, ATA, IDE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,887,145 A (HARARI et al.) 23 MARCH 1999, Fig. 5A. col. 8 lines 40-43.	1-14
X	US 5,802,325 A (LE ROUX) 01 SEPTEMBER 1998, Fig. 2, col. 3 lines 30-34.	1, 2, 6, 7, & 8
A	US 5,630,093 A (HOLZHAMMER et al.) 13 MAY 1997, abstract.	9, 13, & 14
A	US 5,623,637 A (JONES et al.) 22 APRIL 1997, Fig. 3 col. 3 lines 26-29.	1
X,P	US 6,011,741 A (WALLACE et al.) 04 JANUARY 2000, Figs. 13A & 13B col. 7 lines 53-62.	1-14

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 28 JUNE 2000	Date of mailing of the international search report 25 JUL 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer HONG C. KIM Eugenia Logan Telephone No. (703) 305-3900